A low-cost versatile system for continuous real-time respiratory activity measurement as a tool in environmental research

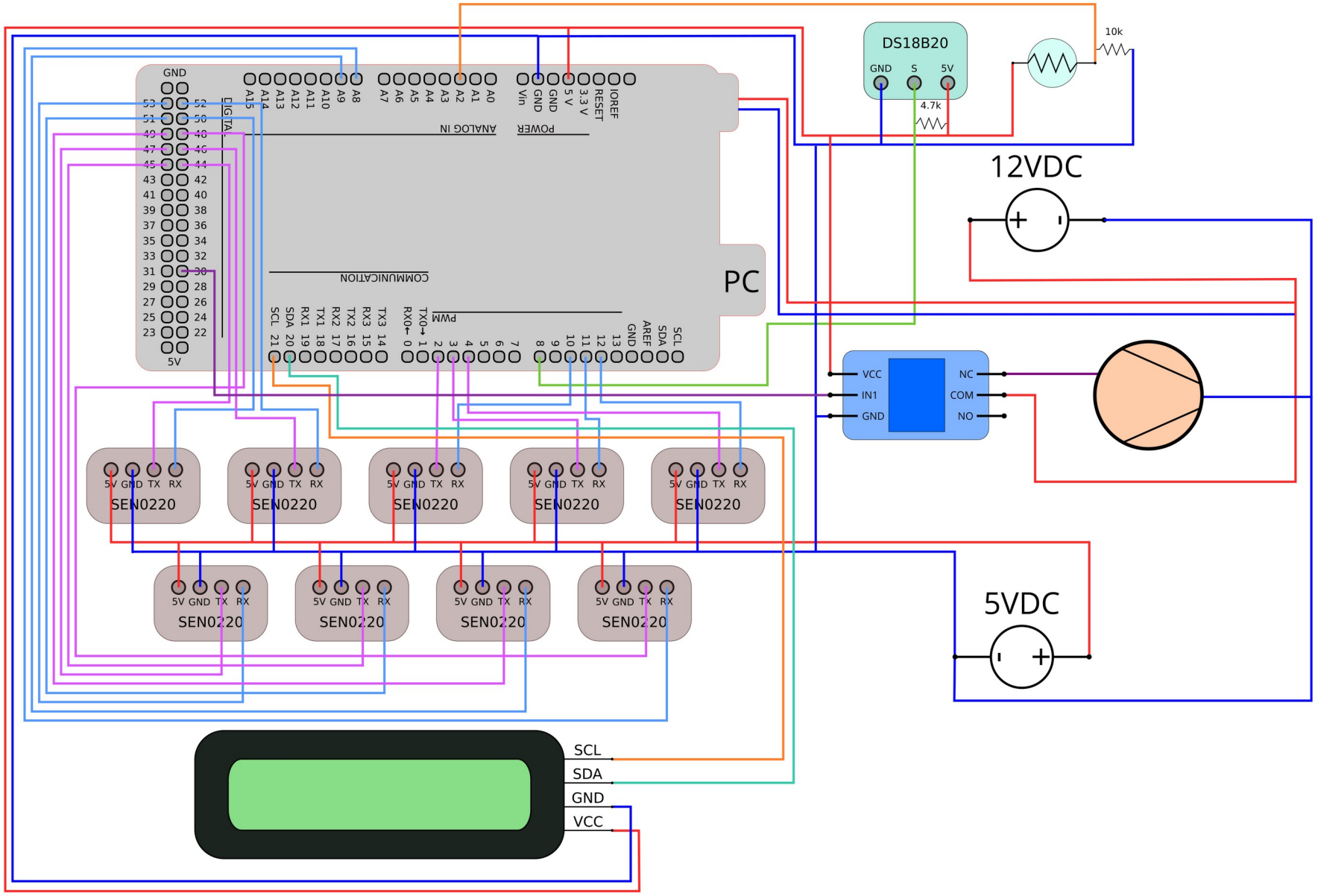
Tamara Djerdj[1], Vesna Peršić[1], Davorka K. Hackenberger[1], Domagoj K. Hackenberger[1], Branimir K. Hackenberger[1,*]

[1] *Josip Juraj Strossmayer University of Osijek, Department of Biology, Cara Hadrijana 8/A, HR-31000 Osijek, Croatia*

[*] *To whom correspondence should be addressed at Josip Juraj Strossmayer University of Osijek, Department of Biology, Cara Hadrijana 8/A, 31000 Osijek, Croatia*

Fax: +385 31 399-939. Phone: +385 31 399-910. E-mail: hackenberger@biologija.unios.hr

Supplementary Material S3. Wiring diagram and Arduino code



```
/* RTOX metabolic rate measuring system software
 *
 * CONNECTING THE COMPONENTS
 *
 * 20x4 LCD screen: VCC - 5V; GND - GND; SDA - D20; SCL - D21
 *
 * SEN0220 NDIR CO2 sensors (Vin - 5V; GND - GND):
 * - Ref (S1): RX1 = 10; TX1 = 2;
 * - Exp1 (S2): RX2 = 11; TX2 = 3;
 * - Exp2 (S3): RX3 = 12; TX3 = 4;
 * - Exp3 (S4): RX4 = 50; TX4 = 44;
 * - Exp4 (S5): RX5 = 51; TX5 = 45;
 * - Exp5 (S6): RX6 = 52; TX6 = 46;
 * - Exp6 (S7): RX7 = 53; TX7 = 47;
 * - Exp7 (S8): RX8 = 62; TX8 = 48;
 * - Exp8 (S9): RX9 = 63; TX9 = 49;
 *
 * DS18B20 temperature sensor: Vin - 5V; GND; signal - D3 (4.7k resistor between 5V and D3)
 *
 * LCD backlight switch button (push-button): D2, GND (2.2k resistor between Data and 5V)
 * Manual reset push-button: 5V, GND, RESET (1k resistor between RESET and 5V)
 * Photosensor: Vin - 5V; GND; A2 (10k resistor between Data and GND)
 *
*/

// Software reset library for Arduino - implementing in cases of sudden CO2 concentration jumps
#include <SoftReset.h>

#include <SoftwareSerial.h>

// LCD screen
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C  lcd(0x27,2,1,0,4,5,6,7,3,POSITIVE); // 0x27 is the I2C bus address for an unmodified backpack

// LCD screen backlight control
const int buttonPin = 6;
volatile int buttonState = 0;
boolean light_on = true;

// Photosensor
int PHS_pin = 2;
float PHS_reading;

// Temperature
#include <DallasTemperature.h>
#include <OneWire.h>

int temp_sensor = 8;
float temp = 0;

OneWire oneWirePin(temp_sensor);
DallasTemperature sensors(&oneWirePin);

// Vacuum-pump control
#define RELAY  30
```

```
int data;

// Preparation of CO2-variables
// Communication with the CO2 sensors
int RX1 = 10; int TX1 = 2;
int RX2 = 11; int TX2 = 3;
int RX3 = 12; int TX3 = 4;
int RX4 = 50; int TX4 = 44;
int RX5 = 51; int TX5 = 45;
int RX6 = 52; int TX6 = 46;
int RX7 = 53; int TX7 = 47;
int RX8 = 62; int TX8 = 48;
int RX9 = 63; int TX9 = 49;

unsigned char hexdata[9] = {0xFF, 0x01, 0x86, 0x00, 0x00, 0x00, 0x00, 0x00, 0x79};

// Storing CO2 values
// Raw data
int CO2_1_r; int CO2_2_r; int CO2_3_r; int CO2_4_r; int CO2_5_r; int CO2_6_r; int CO2_7_r; int CO2_8_r; int
CO2_9_r;
// Average CO2 concentration in t=0
int CO2_0_av;
// Corrected values
int CO2_1_c; int CO2_2_c; int CO2_3_c; int CO2_4_c; int CO2_5_c; int CO2_6_c; int CO2_7_c; int CO2_8_c;
int CO2_9_c;
// CO2 concentration change in unit of time (s)
int v1; int v2; int v3; int v4; int v5; int v6; int v7; int v8; int v9;
// Variables used for correction of raw data
int noise; // deviation from initial value of ambient CO2 concentration
int d_S1; int d_S2; int d_S3; int d_S4; int d_S5; int d_S6; int d_S7; int d_S8; int d_S9; // sensor errors

void setup() {
  Serial.begin(9600);

  pinMode(RELAY, OUTPUT);

  sensors.begin();

  // activate LCD module
  lcd.begin (20,4);
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);

  lcd.home(); // set cursor to 0,0
  lcd.print("   DEFENSoil");
  lcd.setCursor(0,1);
  lcd.print("  RespiroMeter");

  delay(3000);

  lcd.clear();


  CO2_1_r = readCO2(RX1, TX1);
  CO2_2_r = readCO2(RX2, TX2);
  CO2_3_r = readCO2(RX3, TX3);
  CO2_4_r = readCO2(RX4, TX4);
  CO2_5_r = readCO2(RX5, TX5);
  CO2_6_r = readCO2(RX6, TX6);
```

```
  CO2_7_r = readCO2(RX7, TX7);
  CO2_8_r = readCO2(RX8, TX8);
  CO2_9_r = readCO2(RX9, TX9);

  CO2_0_av = (CO2_1_r + CO2_2_r + CO2_3_r + CO2_4_r + CO2_5_r + CO2_6_r + CO2_7_r + CO2_8_r +
CO2_9_r)/9;

  d_S1 = CO2_1_r - CO2_0_av;
  d_S2 = CO2_2_r - CO2_0_av;
  d_S3 = CO2_3_r - CO2_0_av;
  d_S4 = CO2_4_r - CO2_0_av;
  d_S5 = CO2_5_r - CO2_0_av;
  d_S6 = CO2_6_r - CO2_0_av;
  d_S7 = CO2_7_r - CO2_0_av;
  d_S8 = CO2_8_r - CO2_0_av;
  d_S9 = CO2_9_r - CO2_0_av;
}

void loop() {
  CO2_1_r = readCO2(RX1, TX1);
  CO2_2_r = readCO2(RX2, TX2);
  CO2_3_r = readCO2(RX3, TX3);
  CO2_4_r = readCO2(RX4, TX4);
  CO2_5_r = readCO2(RX5, TX5);
  CO2_6_r = readCO2(RX6, TX6);
  CO2_7_r = readCO2(RX7, TX7);
  CO2_8_r = readCO2(RX8, TX8);
  CO2_9_r = readCO2(RX9, TX9);

  noise = CO2_1_r - d_S1 - CO2_0_av;

  CO2_1_c = CO2_1_r - d_S1 - noise;
  CO2_2_c = CO2_2_r - d_S2 - noise;
  CO2_3_c = CO2_3_r - d_S3 - noise;
  CO2_4_c = CO2_4_r - d_S4 - noise;
  CO2_5_c = CO2_5_r - d_S5 - noise;
  CO2_6_c = CO2_6_r - d_S6 - noise;
  CO2_7_c = CO2_7_r - d_S7 - noise;
  CO2_8_c = CO2_8_r - d_S8 - noise;
  CO2_9_c = CO2_9_r - d_S9 - noise;

  PHS_reading = analogRead(PHS_pin);


  sensors.requestTemperatures();
  temp = sensors.getTempCByIndex(0);


//  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("1:");
  lcd.setCursor(4,0);
  lcd.print("     ");
  lcd.setCursor(4,0);
  lcd.print(CO2_2_c);
  lcd.setCursor(11,0);
  lcd.print("2:");
  lcd.setCursor(13,0);
  lcd.print("     ");
```

```
lcd.setCursor(13,0);
lcd.print(CO2_3_c);
lcd.setCursor(2,1);
lcd.print("3:");
lcd.setCursor(4,1);
lcd.print("     ");
lcd.setCursor(4,1);
lcd.print(CO2_4_c);
lcd.setCursor(11,1);
lcd.print("4:");
lcd.setCursor(13,1);
lcd.print("     ");
lcd.setCursor(13,1);
lcd.print(CO2_5_c);
lcd.setCursor(2,2);
lcd.print("5:");
lcd.setCursor(4,2);
lcd.print("     ");
lcd.setCursor(4,2);
lcd.print(CO2_6_c);
lcd.setCursor(11,2);
lcd.print("6:");
lcd.setCursor(13,2);
lcd.print("     ");
lcd.setCursor(13,2);
lcd.print(CO2_7_c);
lcd.setCursor(2,3);
lcd.print("7:");
lcd.setCursor(4,3);
lcd.print("     ");
lcd.setCursor(4,3);
lcd.print(CO2_8_c);
lcd.setCursor(11,3);
lcd.print("8:");
lcd.setCursor(13,3);
lcd.print("     ");
lcd.setCursor(13,3);
lcd.print(CO2_9_c);

if(Serial.available()>0){
  data=Serial.read();
  if(data==1){digitalWrite(RELAY, 1);}
  else if(data==0){digitalWrite(RELAY, 0);}
  else if(data==2){
    Serial.print(CO2_1_r);
    Serial.print(";");
    Serial.print(CO2_1_c);
    Serial.print(";");
    Serial.print(noise);
    Serial.print(";");
    Serial.print(d_S1);
    Serial.print(";");
    Serial.print(CO2_2_r);
    Serial.print(";");
    Serial.print(CO2_2_c);
    Serial.print(";");
    Serial.print(d_S2);
    Serial.print(";");
    Serial.print(CO2_3_r);
```

```
      Serial.print(";");
      Serial.print(CO2_3_c);
      Serial.print(";");
      Serial.print(d_S3);
      Serial.print(";");
      Serial.print(CO2_4_r);
      Serial.print(";");
      Serial.print(CO2_4_c);
      Serial.print(";");
      Serial.print(d_S4);
      Serial.print(";");
      Serial.print(CO2_5_r);
      Serial.print(";");
      Serial.print(CO2_5_c);
      Serial.print(";");
      Serial.print(d_S5);
      Serial.print(";");
      Serial.print(CO2_6_r);
      Serial.print(";");
      Serial.print(CO2_6_c);
      Serial.print(";");
      Serial.print(d_S6);
      Serial.print(";");
      Serial.print(CO2_7_r);
      Serial.print(";");
      Serial.print(CO2_7_c);
      Serial.print(";");
      Serial.print(d_S7);
      Serial.print(";");
      Serial.print(CO2_8_r);
      Serial.print(";");
      Serial.print(CO2_8_c);
      Serial.print(";");
      Serial.print(d_S8);
      Serial.print(";");
      Serial.print(CO2_9_r);
      Serial.print(";");
      Serial.print(CO2_9_c);
      Serial.print(";");
      Serial.print(d_S9);
      Serial.print(";");
      Serial.print(PHS_reading);
      Serial.print(";");
      Serial.println(temp);
    }
  }

  delay(900);
}

int readCO2(int RX, int TX) {
  SoftwareSerial mySerial(RX, TX);

  mySerial.begin(9600);

  mySerial.write(hexdata, 9);
  delay(10);

  for (int i = 0, j = 0; i < 9; i++)
```

```
  {
    if (mySerial.available() > 0)
    {
      long hi, lo, CO2;
      int ch = mySerial.read();

      if (i == 2) {
        hi = ch;    //High concentration
      }
      if (i == 3) {
        lo = ch;    //Low concentration
      }
      if (i == 8) {
        CO2 = hi * 256 + lo; //CO2 concentration
        return(CO2);
      }
    }
  }

  mySerial.end();
}
```